

Texting an LLM shouldn't feel like a monologue



5.02x Average
Reply Ratio

嗯

LLMs are trained to be thorough.

For a companion app,
thoroughness **actively**
destroys the illusion of
conversation.

For short pings, the ratio balloons to 6.60x.
A 1-character text triggers a multi-paragraph essay.

Blunt token limits destroy the illusion of conversation



Hard Max Tokens: Rejected.

Cutting off an LLM mid-thought leaves the user staring at a broken sentence. It breaks immersion harder than any wall of text.

I was thinking about what you
said yesterday and—

System prompts are merely suggestions to chatty personas



The Verbosity Matrix

Mode: Realistic vs. Companion

Relationship:
Distant → Neutral → Close

Persona Profile:
Quiet → Chatty

A quiet persona who just met you should reply with 2-4 characters. A chatty persona in a close relationship can afford 12-18.

Treating length as a single variable guarantees failure.
"Keep it brief" only tames the middle, never the extremes.

Brevity without warmth feels like a cold dismissal



Short vs. Warm Diagnostic Matrix

Short + Cold

嗯。

(2 chars)

Result: Non-answer /
Dismissal (Broken)

Short + Warm

幹嘛，句點我喔？

(8 chars - Playful teasing)

Result: Comfortable
Silence (Natural)

The Minimum Viable Quality Rule

Questions need answers. Greetings need reciprocation.
Every time limits tightened, brevity crossed into brokenness.
Real friends have **comfortable silence**, not coldness.

Combining prompt contracts with deterministic code



The Two-Step Crafting Process

Step 1

Prompt Contract

Dynamic character & bubble budget (Nuance)

Step 2

Budget Box

`normalizeAssistantReplyLength()`
(Hard Limits)



-45% reduction in overall realistic ratio (5.02 → 2.78)

<1ms latency for deterministic code
(vs 500ms+ for secondary LLM calls)

Prompts are good at nuance but bad at hard limits.
Code is good at hard limits but bad at nuance.
Use both to keep the LLM short, warm, and natural.